

Symmetry properties of the Novelli–Pak–Stoyanovskii algorithm

Robin Sulzgruber *

Fakultät für Mathematik, Universität Wien,
Oskar-Morgenstern-Platz 1, A1090 Vienna, Austria.

March 21, 2014

Abstract

The number of standard Young tableaux of a fixed shape is famously given by the hook-length formula due to Frame, Robinson and Thrall. A bijective proof of Novelli, Pak and Stoyanovskii relies on a sorting algorithm akin to jeu-de-taquin which transforms an arbitrary filling of a partition into a standard Young tableau by exchanging adjacent entries. Recently, Krattenthaler and Müller defined the complexity of this algorithm as the average number of performed exchanges, and Neumann and the author proved it fulfils some nice symmetry properties. In this paper we recall and extend the previous results and provide new bijective proofs.

1 Introduction

The main motivation for this work was a conjecture by Krattenthaler and Müller stating that the average runtime of the Novelli–Pak–Stoyanovskii algorithm remains the same whether it is applied row-wise or column-wise. Equivalently, the sorting of all fillings of a partition λ requires the same number of exchanges as the sorting of all fillings of its conjugate λ' .

We consider two additional combinatorial objects, the drop function and exchange numbers, by means of which the complexity can be expressed. The aim of this paper is to provide formulae for the drop function (Theorem 2.2), the exchange numbers (Theorem 2.3) and the complexity (Theorem 2.4). The symmetry in λ and λ' will follow easily from these results (Corollary 2.1).

It should be mentioned that Neumann and the author obtain the symmetry result in a slightly more general setting, defining a family of closely related algorithms and proving that two such algorithms share the same complexity, drop function and exchange numbers if both algorithms produce each standard Young tableau equally often as an

*Supported by the Austrian Science Foundation FWF, grant S50-N15 in the framework of the Special Research Program “Algorithmic and Enumerative Combinatorics” (SFB F50).

Email: robin.sulzgruber@univie.ac.at

output. However, restricting ourselves to the original algorithm, we are able to deduce more explicit results.

Finally, some of these new formulae generalise the hook-length formula, and demand bijective proofs. We succeed in providing bijections for the drop function (Theorem 3.1) and the exchange numbers (Proposition 3.1), and give a bijective version of the symmetry of the complexity at the end of Section 3.

In the rest of this section we define the combinatorial objects we are going to consider, and recall the Novelli–Pak–Stoyanovskii algorithm.

Let $n \in \mathbb{N}$. A partition of n is a sequence $\lambda = (\lambda_1, \lambda_2, \dots)$ of non-negative integers such that $\lambda_i \geq \lambda_{i+1}$ and $\sum \lambda_i = n$. If this is the case we write $\lambda \vdash n$. For us the appropriate way to represent a partition is via its Young diagram $\lambda = \{(i, j) \in \mathbb{Z}^2 : 1 \leq i, 1 \leq j \leq \lambda_i\}$. The elements $x = (i, j) \in \lambda$ are called the cells of the partition. We visualise a partition as a left-justified array with λ_i cells in the i -th row. Thus, the cells are arranged like the entries of a matrix (see Figure 1). The conjugated partition of λ is defined as

$$\lambda' = \{(i, j) \in \mathbb{Z}^2 : 1 \leq i, 1 \leq j \leq \lambda'_i\} := \{(i, j) : (j, i) \in \lambda\}.$$

Given a cell $x = (i, j)$ in λ we define its arm $\text{arm}_\lambda(x) := \lambda_i - j$ as the number of cells to the right of x . The leg $\text{leg}_\lambda(x) := \lambda'_j - i$ is defined as the number of cells below x . Furthermore, we define the hook-length of x as $h_\lambda(x) := \text{arm}_\lambda(x) + \text{leg}_\lambda(x) + 1$, and the cohook-length as $h'(x) := i + j - 2$. Finally, we denote the set of top and left neighbours of x by $N_\lambda^-(x) := \{(i-1, j), (i, j-1)\} \cap \lambda$ and the set of right and bottom neighbours of x by $N_\lambda^+(x) := \{(i, j+1), (i+1, j)\} \cap \lambda$.

We want to consider integer fillings of partitions, i.e. maps $T : \lambda \rightarrow \mathbb{Z}$. The image $k = T(x)$ is called the entry of the cell x . The partition λ is also called the shape of the filling. A tabloid is a bijection $T : \lambda \rightarrow \{1, \dots, n\}$, i.e. a filling where each entry between 1 and n occurs exactly once. We say a tabloid T is sorted at x if $T(x)$ is less than all entries among the bottom and right neighbours of x . A tabloid which is sorted at every cell is called a standard Young tableau. Finally, a hook tableau is a map $H : \lambda \rightarrow \mathbb{Z}$ such that $-\text{leg}_\lambda(x) \leq H(x) \leq \text{arm}_\lambda(x)$ for all $x \in \lambda$.

Let $\text{T}(\lambda)$ denote the set of all tabloids, $\text{SYT}(\lambda)$ the set of all standard Young tableaux and $\text{H}(\lambda)$ the set of all hook tableaux of shape λ . It is obvious that

$$\#\text{T}(\lambda) = n! \quad \text{and} \quad \#\text{H}(\lambda) = \prod_{x \in \lambda} h_\lambda(x). \quad (1)$$

The symmetric group \mathfrak{S}_n acts simply transitively on $\text{T}(\lambda)$ via $\sigma(T) := \sigma \circ T$. Let $T \in \text{T}(\lambda)$ and $\sigma = (k_0, \dots, k_r)$ be a cycle such that $T^{-1}(k_i)$ is a right or bottom neighbour of $T^{-1}(k_{i-1})$ for all $1 \leq i \leq r$. Then σ is called a forward slide on T at x if $T(x) = k_0$, $k_i < k_{i+1}$ for all $1 \leq i < r$ and $k_0 > k_r$. If instead $k_0 = T(x)$ and $k_i < k_{i+1}$ for all $0 \leq i < r$ then the inverse σ^{-1} is called a backward slide on T at x . For example $(7, 1, 2, 5)$ is a forward slide on the (left) tabloid in Figure 1 while $(1, 2, 8)^{-1}$ is a backward slide.

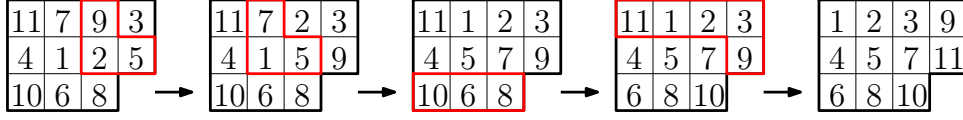


Figure 2: The non-trivial forward slides are $(9, 2, 5)$, $(7, 1, 5)$, $(10, 6, 8)$ and $(11, 1, 2, 3, 9)$.

The inverse algorithm is given as a sequence of backward slides, however, to determine the correct backward slide at each step, the hook tableau has to be taken into account. For details see [NPS97, Sag01].

We remark that Neumann and the author define analogous sorting algorithms in [NS13], by replacing \prec with a more general order \prec_U . This order is induced by an arbitrary standard Young tableaux U of the same shape as T , and defined by $x \prec_U y$ if and only if $U(x) < U(y)$.

The order \prec is special in the following sense: If we sort all tabloids in $T(\lambda)$ with respect to \prec then each standard Young tableau in $\text{SYT}(\lambda)$ is met exactly $\prod_{x \in \lambda} h_\lambda(x)$ times. If we sort with respect to \prec_U for an arbitrary standard Young tableau U , this need not be the case anymore. Interestingly, it turns out that if two algorithms arising from different orders, produce each standard Young tableau equally often as an output, then they must automatically share a lot of properties [NS13, Corollary 4.5, Corollary 5.4]. In this paper we restrict ourselves to the order \prec and improve the previous general results for this special case.

The following property of the Novelli–Pak–Stoyanovskii algorithm is so useful that it deserves its own lemma.

Lemma 1.1 (Invariance) *Let $k, n \in \mathbb{N}$ such that $k \leq n$, let $\lambda \vdash n$ be a partition, $\pi \in \mathfrak{S}_n$ be a permutation that fixes all l where $1 \leq l < k$, and $T \in T(\lambda)$ be a tabloid. Set $\tilde{T} := \pi \circ T$ and let T_x, \tilde{T}_x be the tabloids which arise during the sorting of T and \tilde{T} , respectively. Then each permutation π_x defined by $\tilde{T}_x = \pi_x \circ T_x$ fixes all l with $1 \leq l < k$.*

Proof. We apply induction on x with respect to \prec . The claim is trivial for the maximal cell. Thus, let $x \in \lambda$ be a cell with successor $y \succ x$, and suppose that π_y fixes all l with $l < k$.

Let $\sigma_x = (a_0, a_1, \dots, a_r)$ and $\tilde{\sigma}_x = (b_0, b_1, \dots, b_s)$ be the maximal forward slides defined as above, such that $a_0 = T_y(x)$ and $b_0 = \tilde{T}_y(x)$. Since we always prefer smaller entries over larger ones, and since all entries l with $l < k$ appear in the same positions in T_y and \tilde{T}_y , whenever we choose an entry $a_i < k$ we must make the same choice for b_i . Thus, we may decompose σ_x and $\tilde{\sigma}_x$ into forward slides

$$\sigma_x = \pi_1 \circ (a_0, a_1, \dots, a_t) \quad \text{and} \quad \tilde{\sigma}_x = \pi_2 \circ (b_0, a_1, \dots, a_t),$$

such that $a_t < k$ and where π_1 and π_2 fix all l with $l < k$. If $a_0 < k$ then $a_0 = b_0$ and $\sigma_x = \tilde{\sigma}_x$ fixes all l with $l \geq k$. Hence,

$$\tilde{T}_x = \sigma_x \circ \tilde{T}_y = \sigma_x \pi_y \circ T_y = \pi_y \sigma_x \circ T_y = \pi_y \circ T_x.$$

On the other hand, if $a_0 \geq k$ then also $b_0 \geq k$. Write $\pi_y = \pi_3 \circ (a_0, b_0, c_1, \dots, c_p)$ such that π_3 fixes a_0, b_0 and all l with $l < k$. Then we compute

$$\begin{aligned}\tilde{T}_x &= \pi_2(b_0, a_1, \dots, a_t) \circ \tilde{T}_y \\ &= \pi_2(b_0, a_1, \dots, a_t) \pi_3(a_0, b_0, c_1, \dots, c_p) \circ T_y \\ &= \pi_2 \pi_3(b_0, a_1, \dots, a_t)(a_0, b_0, c_1, \dots, c_p) \circ T_y \\ &= \pi_2 \pi_3(a_0, b_0, c_1, \dots, c_p)(a_0, a_1, \dots, a_t) \circ T_y \\ &= \pi_2 \pi_y \pi_1^{-1} \pi_1(a_0, a_1, \dots, a_t) \circ T_y = \pi_2 \pi_y \pi_1^{-1} \circ T_x.\end{aligned}$$

Thus, we have $\pi_x = \pi_y$ in the first case and $\pi_x = \pi_2 \pi_y \pi_1^{-1}$ in the second case. ■

Krattenthaler and Müller were interested in the average number of steps needed to sort a tabloid of a given shape. Therefore, fix $T \in \mathbf{T}(\lambda)$ and let $r(T, x)$ be the length of the forward slide σ_x . As σ_x can be decomposed into $r(T, x)$ exchanges, the total number of exchanges needed to transform T into a standard Young tableau is

$$r(T) := \sum_{x \in \lambda} r(T, x).$$

The average number of needed exchanges is

$$C(\lambda) := \frac{1}{n!} \sum_{T \in \mathbf{T}(\lambda)} r(T).$$

We call the number $C(\lambda)$ the complexity of the Novelli–Pak–Stoyanovskii algorithm on the shape λ . The surprising observation by Krattenthaler and Müller can now be stated as $C(\lambda) = C(\lambda')$. In [NS13] two related objects are considered. The first of these is the drop function which counts the number of tabloids in which the entry k , where $1 \leq k \leq n$, is moved to the cell $x \in \lambda$ by its own forward slide. We denote the drop function by

$$d_\lambda(k, x) := \#\{T \in \mathbf{T}(\lambda) : \text{there exists } z \in \lambda \text{ such that } T(z) = k, T_z(x) = k\}.$$

The second kind of objects are exchange numbers which count how often two entries are exchanged. That is, given entries k and l , where $1 \leq k < l \leq n$, we are interested in the number of tabloids such that the exchange (k, l) appears in the decomposition of a maximal forward slide σ_z into exchanges, for some cell $z \in \lambda$. Formally, we define the exchange numbers as

$$\varepsilon_\lambda(k, l) := \#\{T \in \mathbf{T}(\lambda) : k < l \text{ and there exists } z \in \lambda \text{ such that } T(z) = l, \sigma_z(k) \neq k\}.$$

For example, in Figure 2 the entries 5, 9 and 11 all drop to the cell $(2, 4)$. The entry 1 is exchanged with the entries 7 and 11, and the tabloid contributes ten exchanges to the overall complexity $C(\lambda)$, where $\lambda = (4, 4, 3)$.

2 Formulae

In this section we start out with some preparations we need to define signed exit numbers. Afterwards, we prove formulae for the complexity, the drop function and the exchange numbers, which imply their symmetry in λ and λ' . We shall derive them from Theorem 2.1 which is a result on signed exit numbers.

Let k and l be two entries such that $1 \leq k < l \leq n$ and $x, y \in \lambda$ two cells. We are interested in the number of times k and l are exchanged at the positions x and y during the sorting of all tabloids. Suppose that $T^{-1}(l) = z$, then this happens if and only if σ_z moves the entry k from the cell x to the cell y . To make this precise, let $z_1 \prec \dots \prec z_n$ be the cells of λ . We define the local exchange numbers as

$$\varepsilon_\lambda(k, l, x, y) := \#\{T \in \mathsf{T}(\lambda) : k < l \text{ and } T(z_i) = l, T_{z_{i+1}}(x) = k, T_{z_i}(y) = k \text{ for some } i\}.$$

It is immediate from their definition that $\varepsilon(k, l, x, y) = 0$ unless x and y are neighbouring cells. More precisely, since $k < l$ the cell x must be a bottom of right neighbour of y . An interesting property of local exchange numbers is that they do not depend on the larger argument l .

Lemma 2.1 *Let $k, l, m, n \in \mathbb{N}$ such that $1 \leq k < l, m \leq n$, let $\lambda \vdash n$ be a partition and $x, y \in \lambda$ be two cells. Then*

$$\varepsilon_\lambda(k, l, x, y) = \varepsilon_\lambda(k, m, x, y).$$

It is therefore convenient to define $\varepsilon_\lambda(k, x, y) := \varepsilon_\lambda(k, n, x, y)$ and $\varepsilon_\lambda(k) := \varepsilon_\lambda(k, n)$. This lemma is proven in [NS13, Proposition 4.1]. We shall give a short proof using the Invariance Lemma 1.1.

Proof. Let $T \in \mathsf{T}(\lambda)$, and set $\tilde{T} := (l, m) \circ T$. Let $z := T^{-1}(l) = \tilde{T}^{-1}(m)$. By Lemma 1.1 the forward slide σ_z moves k from x to y during the sorting of T if and only if $\tilde{\sigma}_z$ does the same during the sorting of \tilde{T} . Thus, the map $T \mapsto (l, m) \circ T$ provides an involution on $\mathsf{T}(\lambda)$ mapping tabloids which contributing to $\varepsilon_\lambda(k, l, x, y)$ to those which contribute to $\varepsilon_\lambda(k, m, x, y)$. \blacksquare

Following [NS13], we define the signed exit numbers as

$$\Delta_\lambda(k, x) := \sum_{y \in N_\lambda^-(x)} \varepsilon_\lambda(k, x, y) - \sum_{y \in N_\lambda^+(x)} \varepsilon_\lambda(k, y, x).$$

While this definition is not very intuitive at first glance, the signed exit numbers allow for a unified approach to the exchange numbers and the drop function. For $k \in \mathbb{N}$ and a cell $x \in \lambda$ we define

$$f_\lambda(k, x) := \#\{T \in \mathsf{SYT}(\lambda) : T(x) = k\}.$$

Theorem 2.1 (Signed Exit Numbers) *Let $k, n \in \mathbb{N}$ such that $1 \leq k < n$, let $\lambda \vdash n$ be a partition and $x \in \lambda$ be a cell. Then we have the recursion*

$$(n - k) \Delta_\lambda(k, x) = (n - 1)! - \frac{n! f_\lambda(k, x)}{f_\lambda} + \sum_{l=1}^{k-1} \Delta_\lambda(l, x). \quad (4)$$

Solving the recursion we obtain

$$\Delta_\lambda(k, x) = \frac{n!}{(n - k)(n - k + 1)} \left(1 - \frac{f_\lambda(k, x)}{f_\lambda} (n - k + 1) - \sum_{l=1}^{k-1} \frac{f_\lambda(l, x)}{f_\lambda} \right). \quad (5)$$

Proof. The recursion (4) was deduced in [NS13, Theorem 5.3] in a slightly more general setting. Its proof makes use of Lemma 2.1.

One can derive (5) from (4) by induction on k where the base of the induction and the induction step are both covered by the same computation:

$$\begin{aligned} \Delta_\lambda(k, x) &= \frac{(n - 1)!}{n - k} - \frac{n! f_\lambda(k, x)}{(n - k) f_\lambda} + \frac{1}{n - k} \sum_{l=1}^{k-1} \Delta_\lambda(l, x) \\ &= \frac{(n - 1)!}{n - k} - \frac{n! f_\lambda(k, x)}{(n - k) f_\lambda} + \frac{n!}{n - k} \sum_{l=1}^{k-1} \frac{1}{(n - l)(n - l + 1)} \\ &\quad - \sum_{l=1}^{k-1} \frac{n!}{(n - k)(n - l)} \frac{f_\lambda(l, x)}{f_\lambda} - \sum_{l=1}^{k-1} \sum_{i=1}^{l-1} \frac{n!}{(n - k)(n - l)(n - l + 1)} \frac{f_\lambda(i, x)}{f_\lambda} \\ &= \frac{(n - 1)!}{n - k} - \frac{n! f_\lambda(k, x)}{(n - k) f_\lambda} + \frac{n!}{n - k} \left(\frac{1}{n - k + 1} - \frac{1}{n} \right) \\ &\quad - \sum_{l=1}^{k-1} \frac{n!}{(n - k)(n - l)} \frac{f_\lambda(l, x)}{f_\lambda} - \sum_{i=1}^{k-2} \sum_{l=i+1}^{k-1} \frac{n!}{(n - k)(n - l)(n - l + 1)} \frac{f_\lambda(i, x)}{f_\lambda} \\ &= \frac{n!}{(n - k)(n - k + 1)} - \frac{n! f_\lambda(k, x)}{(n - k) f_\lambda} - \sum_{l=1}^{k-1} \frac{n!}{(n - k)(n - l)} \frac{f_\lambda(l, x)}{f_\lambda} \\ &\quad - \sum_{i=1}^{k-2} \frac{n!}{n - k} \frac{f_\lambda(i, x)}{f_\lambda} \left(\frac{1}{n - k + 1} - \frac{1}{n - i} \right) \\ &= \frac{n!}{(n - k)(n - k + 1)} \left(1 - \frac{f_\lambda(k, x)}{f_\lambda} (n - k + 1) - \sum_{l=1}^{k-1} \frac{f_\lambda(l, x)}{f_\lambda} \right) \end{aligned}$$

■

Note that the recursion (4) cannot be solved in the setting of [NS13]. In contrast to the more general approach we obtain the following formula for the drop function.

Theorem 2.2 (Drop Function) *Let $k, n \in \mathbb{N}$ such that $1 \leq k \leq n$, let $\lambda \vdash n$ be a partition and $x \in \lambda$ be a cell. Then*

$$d_\lambda(k, x) = \frac{n!}{n - k + 1} \left(1 - \sum_{l=1}^{k-1} \frac{f_\lambda(l, x)}{f_\lambda} \right) \quad (6)$$

$$= \frac{\prod_{y \in \lambda} h_\lambda(y)}{n - k + 1} \sum_{l=k}^n f_\lambda(l, x). \quad (7)$$

Proof. An entry k drops to x if it starts there or is moved there by an exchange with a smaller entry and is not moved away by an exchange with a smaller entry. Thus,

$$d_\lambda(k, x) = (n - 1)! + \sum_{l=1}^{k-1} \Delta_\lambda(l, x) = (n - k) \Delta_\lambda(k, x) + \frac{n! f_\lambda(k, x)}{f_\lambda}. \quad (8)$$

Substituting (5) in (8) yields (6). Now, (7) follows from (6) using $\sum_{k=1}^n f_\lambda(k, x) = f_\lambda$ and the hook-length formula (2). \blacksquare

Note that (7) can be viewed as a generalised hook-length formula. In the case $n = 1$ it reduces to

$$n! = n(n - 1)! = n d_\lambda(1, x) = \frac{n}{n} \prod_{y \in \lambda} h_\lambda(y) \sum_{l=1}^n f_\lambda(l, x) = \prod_{y \in \lambda} h_\lambda(y) \cdot \# \text{SYT}(\lambda).$$

Next, we obtain a description of the exchange numbers.

Theorem 2.3 (Exchange Numbers) *Let $k, n \in \mathbb{N}$ such that $1 \leq k < n$, and $\lambda \vdash n$ be a partition. Then we have the recursion*

$$(n - k) \varepsilon_\lambda(k) = (n - 1)! \sum_{x \in \lambda} h'(x) + \sum_{l=1}^{k-1} \varepsilon_\lambda(l) - \frac{n!}{f_\lambda} \sum_{x \in \lambda} h'(x) f_\lambda(k, x). \quad (9)$$

Furthermore,

$$\varepsilon_\lambda(k) = \frac{n!}{(n - k)(n - k + 1)} \sum_{x \in \lambda} h'(x) \left(1 - \frac{f_\lambda(k, x)}{f_\lambda} (n - k + 1) - \sum_{l=1}^{k-1} \frac{f_\lambda(l, x)}{f_\lambda} \right). \quad (10)$$

Proof. The recursion was deduced in [NS13, Theorem 4.4] (again in a slightly more general setting) but both claims now follow directly from Theorem 2.1 and the fact that

$$\varepsilon_\lambda(k) = \sum_{x \in \lambda} \sum_{y \in N_\lambda^-(x)} \varepsilon_\lambda(k, x, y) = \sum_{x \in \lambda} h'(x) \Delta_\lambda(k, x).$$

\blacksquare

For $n \in \mathbb{N}$ let $H_n := \sum_{k=1}^n \frac{1}{k}$ denote the n -th harmonic number.

Theorem 2.4 (Complexity) *Let $n \in \mathbb{N}$ and $\lambda \vdash n$ be a partition. Then we have*

$$C(\lambda) = \left(\sum_{i \geq 1} \binom{\lambda_i}{2} + \sum_{i \geq 1} \binom{\lambda'_i}{2} \right) (H_n - 1) - \sum_{x \in \lambda} \sum_{k=1}^{n-1} h'(x) \frac{f_\lambda(k, x)}{f_\lambda} H_{n-k} \quad (11)$$

$$= \sum_{x \in \lambda} \sum_{k=1}^{n-1} h'(x) \frac{f_\lambda(k, x)}{f_\lambda} (H_n - H_{n-k} - 1). \quad (12)$$

Proof. On the one hand, the complexity is given in terms of exchange numbers as

$$C(\lambda) = \frac{1}{n!} \sum_{k=1}^{n-1} (n-k) \varepsilon_\lambda(k). \quad (13)$$

On the other hand, the complexity can be expressed in terms of the drop function as

$$C(\lambda) = \frac{1}{n!} \sum_{x \in \lambda} \sum_{k=1}^n h'(x) (d_\lambda(k, x) - (n-1)!) \quad (14)$$

$$= \frac{1}{n!} \sum_{x \in \lambda} \sum_{k=1}^n h'(x) \left(d_\lambda(k, x) - \frac{n! f_\lambda(k, x)}{f_\lambda} \right), \quad (15)$$

where (14) counts how often an entry is exchanged with a smaller one (i.e., it drops), and (15) counts how often an entry is exchanged with a larger one.

There are various possibilities to prove the claims. Substitution of either (10) in (13) or (6) in (14) yields (11). Inserting (7) into (15) we obtain (12). Moreover, (11) can easily be transformed into (12) and vice versa using

$$\sum_{x \in \lambda} \sum_{k=1}^n h'(x) \frac{f_\lambda(k, x)}{f_\lambda} = \sum_{x \in \lambda} h'(x) = \sum_{i \geq 1} \binom{\lambda_i}{2} + \sum_{i \geq 1} \binom{\lambda'_i}{2}.$$

■

As promised, the symmetry conjecture is now an immediate consequence. For any cell $x = (i, j) \in \lambda$ set $x' := (j, i) \in \lambda'$. For any tabloid $T \in \mathcal{T}(\lambda)$ define a tabloid $T' \in \mathcal{T}(\lambda')$ by letting $T'(x') := T(x)$. For any hook tableau $H \in \mathcal{H}(\lambda)$ define a hook tableau $H' \in \mathcal{H}(\lambda')$ by $H'(x') := -H(x)$. The induced maps $\lambda \rightarrow \lambda'$, $\mathcal{T}(\lambda) \rightarrow \mathcal{T}(\lambda')$ and $\mathcal{H}(\lambda) \rightarrow \mathcal{H}(\lambda')$ are obviously bijective and involutive in the sense that $(x')' = x$, $(T')' = T$, and $(H')' = H$. Moreover, $T \in \text{SYT}(\lambda)$ if and only if $T' \in \text{SYT}(\lambda')$.

Corollary 2.1 (Symmetry) *Let $k, n \in \mathbb{N}$ such that $1 \leq k \leq n$, let $\lambda \vdash n$ be a partition and $x \in \lambda$ be a cell. Then*

$$C(\lambda) = C(\lambda'), \quad d_\lambda(k, x) = d_{\lambda'}(k, x'), \quad \varepsilon_\lambda(k) = \varepsilon_{\lambda'}(k), \quad \text{and} \quad \Delta_\lambda(k, x) = \Delta_{\lambda'}(k, x').$$

Proof. The assertions follow from Theorems 2.1, 2.2, 2.3 and 2.4 and the fact that $f_\lambda(k, x) = f_{\lambda'}(k, x')$ by virtue of the bijection $T \mapsto T'$. \blacksquare

However, note that $\varepsilon_\lambda(k, x, y) = \varepsilon_{\lambda'}(k, x', y')$ is false in general.

In the rest of this section we explain how the complexity $C(\lambda)$ of the Novelli–Pak–Stoyanovskii algorithm on a given shape $\lambda \vdash n$ can actually be computed.

The most primitive approach is certainly to sort all $n!$ tabloids, counting the exchanges. Theorem 2.4 allows for a somewhat better method. That is, now we “only” need to consider all standard Young tableaux of shape λ , and count how often each entry occurs in each cell, in order to get a hold on the numbers $f_\lambda(k, x)$. In some special cases these numbers can be given a closed form. In general the cost of computing $f_\lambda(k, x)$ “only” depends on the number of sub-partitions of λ . To see this, decompose a tabloid T with $T(x) = k$ into the parts U with entries smaller than k and V with entries larger than k . Then U is a standard Young tableau of shape μ for some $\mu \subseteq \lambda$ while V corresponds to a skew standard Young tableaux. Hence, the numbers $f_\lambda(k, x)$ can be computed by counting standard Young tableaux and skew tableaux whose shape depends on a suitable μ . This can be done using the hook-length formula (2) for standard Young tableaux and Aitken’s determinant formula [Ait43] for the skew tableaux.

3 Bijections

We are now going to give bijective proofs of the formulae and symmetry results in the last section.

Let $T(\lambda, k \rightarrow x)$ denote the set of all tabloids of shape λ such that the entry k drops to the cell x during the application of the sorting algorithm. Moreover, let $\text{SYT}(\lambda, x \geq k)$ be the set of standard Young tableaux of shape λ such that the entry of the cell x is at least k . Theorem 2.2 (7) suggests we should look for a bijection

$$\Psi : T(\lambda, k \rightarrow x) \times \{k, \dots, n\} \rightarrow H(\lambda) \times \text{SYT}(\lambda, x \geq k).$$

Indeed, given a tabloid $T \in T(\lambda, k \rightarrow x)$ we notice that the bijection (3) maps T to a pair $\Phi(T) = (H, U)$ of a hook tableau H and a standard Young tableau $U \in \text{SYT}(\lambda, x \geq k)$. This is true because k drops to the cell x , and can only be moved away by a larger entry afterwards. Now, consider a pair (T, l) of a tabloid $T \in T(\lambda, k \rightarrow x)$ and an integer $k \leq l \leq n$. If we apply Φ to the tabloid $(kl) \circ T$ then by the Invariance Lemma 1.1 we will again obtain a hook tableau H and a standard Young tableau $U \in \text{SYT}(\lambda, x \geq k)$.

Theorem 3.1 *Let $k, n \in \mathbb{N}$ such that $1 \leq k \leq n$, $\lambda \vdash n$ be a partition and $x \in \lambda$ be a cell. Then the map*

$$\begin{aligned} \Psi : T(\lambda, k \rightarrow x) \times \{k, \dots, n\} &\rightarrow H(\lambda) \times \text{SYT}(\lambda, x \geq k), \\ (T, l) &\mapsto \Phi((kl) \circ T), \end{aligned}$$

where Φ is given by the Novelli–Pak–Stoyanovskii algorithm, is a bijection.

Proof. The remarks above show that Ψ is well-defined. We prove the claim by constructing an inverse map. Therefore, let $(H, U) \in \mathcal{H}(\lambda) \times \text{SYT}(\lambda, x \geq k)$. Since the map $\Phi : \mathcal{T}(\lambda) \rightarrow \mathcal{H}(\lambda) \times \text{SYT}(\lambda)$ is a bijection, there is a well-defined tabloid $T := \Phi^{-1}(H, U)$. If $T \in \mathcal{T}(\lambda, k \rightarrow x)$ then set $\Psi^{-1}(H, U) := (T, k)$. In this case we have $\Psi^{-1} \circ \Psi(T, k) = (T, k)$ and $\Psi \circ \Psi^{-1}(H, U) = (H, U)$.

If $T \notin \mathcal{T}(\lambda, k \rightarrow x)$ then we have to determine an integer l , where $k < l \leq n$, such that k drops to x during the sorting of $(kl) \circ T$. To this end, let $I(x) \subseteq \lambda$ be the set of cells which lie to the right and below x . That is, $z \in I(x)$ if and only if $z \in \lambda$ and $z \geq x$ with respect to the component-wise order.

We consider the tabloid T_x . If $T_x(z) \geq k$ for all cells $z \in I(x)$ then we set $l := T(x)$, and claim that $(kl) \circ T \in \mathcal{T}(\lambda, k \rightarrow x)$. To see this, let $\tilde{T} := (kl) \circ T$. By the Invariance Lemma 1.1 we have $\tilde{T}_x(z) \geq k$ for all $z \in I(x)$. Since \tilde{T}_x is sorted on $I(x)$ it follows that $\tilde{T}_x(z) > k$ for all $z \in I(x) \setminus \{x\}$ and $\tilde{T}_x(x) = k$. But this means that the entry k has not moved at all but dropped to its initial cell x .

Otherwise, there is a $z \in I(x)$ such that $T_x(z) < k$. Indeed, we must have $T_x(x) < k$ since T_x is sorted on $I(x)$. In this case let y be the maximal cell with respect to the order \prec used in the Novelli–Pak–Stoyanovskii algorithm such that $T_y(z) \geq k$ for all $z \in I(x)$. Such a y must exist because $U(x) \geq k$. We set $l := T(y)$ and claim that $\tilde{T} := (kl) \circ T \in \mathcal{T}(\lambda, k \rightarrow x)$ as before. By the Invariance Lemma 1.1, y is the maximal cell with respect to \prec such that $\tilde{T}_y(z) \geq k$ for all $z \in I(x)$. This means that the entry $\tilde{T}(y)$ must drop to a cell in $I(x)$. But $\tilde{T}(y) = k$ is the smallest entry of \tilde{T}_y among the cells of $I(x)$. Since \tilde{T}_y is ordered on $I(x)$, the entry k must drop to the cell x .

Once we have found a suitable l as described above, we set $\Psi^{-1}(H, U) := ((kl) \circ T, l)$. Clearly, we have $\Psi \circ \Psi^{-1}(H, U) = (H, U)$.

Conversely, let $T \in \mathcal{T}(\lambda, k \rightarrow x)$ and $k < l \leq n$. Set $\tilde{T} := (kl) \circ T$ and $y := T^{-1}(k) = \tilde{T}^{-1}(l)$. Then $y = x$ or y is the maximal cell of λ with respect to \prec such that $\tilde{T}_y(z) \geq k$ for all $z \in I(x)$. This once again follows from the Invariance Lemma 1.1.

Hence, $\Psi^{-1} \circ \Psi(T, l) = (T, l)$, and the proof is complete. \blacksquare

Note that Ψ induces a bijection

$$\begin{aligned} \mathcal{T}(\lambda, k \rightarrow x) \times \{k, \dots, n\} &\rightarrow \mathcal{T}(\lambda', k \rightarrow x') \times \{k, \dots, n\}, \\ (T, l) &\xrightarrow{\Psi_\lambda} (H, U) \mapsto (H', U') \xrightarrow{\Psi_{\lambda'}^{-1}} (V, m), \end{aligned} \tag{16}$$

which is an involution in the obvious sense. Unfortunately, it does not seem to be easy to eliminate the auxiliary integers. As Figure 3 shows, even in simple cases distinct tabloids may be mapped to the same tabloid with different integer labels.

Using the ideas from Theorem 3.1, we want to find a bijective proof of the formula (10) for the number of exchanges of k with a larger entry. Suppose during the sorting of the tabloid $T \in \mathcal{T}(\lambda)$ the entries k and l with $k < l$ are exchanged. We identify this exchange with the pair (T, l) . Let $\text{Ex}(\lambda, k)$ denote the set of all such exchanges, i.e. $\#\text{Ex}(\lambda, k) = (n - k) \varepsilon_\lambda(k)$.

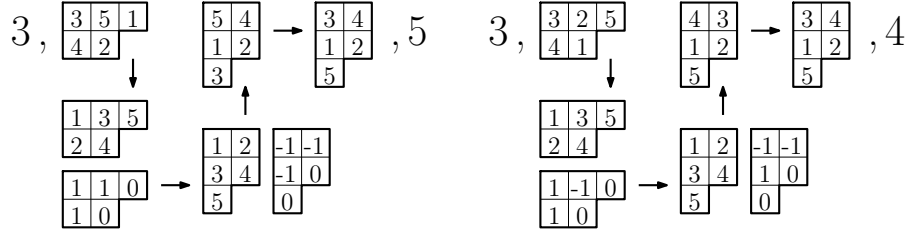


Figure 3: Two examples where $\lambda = (3, 2)$, $k = 3$ and $x = (1, 2)$.

Now, fix a tabloid U . Suppose that k drops to x during the sorting of U , and is afterwards moved to the final cell y . Clearly,

$$\#\{(T, l) \in \text{Ex}(\lambda, k) : T = U\} = h'(x) - h'(y).$$

We denote this cardinality by $e(U, k)$. Thus, we obtain a bijection

$$\text{Ex}(\lambda, k) \rightarrow \{(T, i) : T \in \text{T}(\lambda), 1 \leq i \leq e(T, k)\},$$

for example by ordering the exchanges (T, l_i) , $1 \leq i \leq e(T, k)$ with respect to the order in which they occur during the sorting. Thereby, letting

$$A(\lambda, k) := \bigcup_{x \in \lambda} \text{T}(\lambda, k \rightarrow x) \times \{1, \dots, h'(x)\},$$

we have an inclusion $\iota : \text{Ex}(\lambda, k) \rightarrow A(\lambda, k)$. Moreover, set

$$B(\lambda, k) := \text{H}(\lambda) \times \bigcup_{y \in \lambda} \{T \in \text{SYT}(\lambda) : T(y) = k\} \times \{1, \dots, h'(y)\}.$$

Proposition 3.1 *Let $k, n \in \mathbb{N}$ such that $1 \leq k \leq n$, and $\lambda \vdash n$ be a partition. Then the map*

$$\psi : A(\lambda, k) \setminus \text{Ex}(\lambda, k) \rightarrow B(\lambda, k)$$

defined by $(T, i) \mapsto (H, U, i - e(T, k))$, where $(H, U) := \Phi(T)$ is given by the Novelli–Pak–Stoyanovskii algorithm, is a bijection.

Proof. Let (T, i) be a pair of a tabloid and an integer, and let x be the cell to which k drops during the sorting of T . Set $(H, U) := \Phi(T)$ and $y := U^{-1}(k)$. Then the inequalities

$$e(T, k) = h'(x) - h'(y) < i \leq h'(x) \quad \text{and} \quad 0 < i - e(T, k) \leq h'(y)$$

are equivalent. Thus, (T, i) lies in $A(\lambda, k)$ but not in $\text{Ex}(\lambda, k)$ if and only if $(H, U, i - e(T, k))$ lies in $B(\lambda, k)$. It follows that Ψ is well defined and a bijection. \blacksquare

Considering only cardinalities, where we use Theorem 3.1 to specify the cardinality of $A(\lambda, k)$, Proposition 3.1 reduces to (10). Lastly, we want to give a bijective proof of $C(\lambda) = C(\lambda')$. We do this by indicating a bijection

$$\tilde{\psi} : \text{Ex}(\lambda, k) \times \{k, \dots, n\} \rightarrow \text{Ex}(\lambda', k) \times \{k, \dots, n\}.$$

The bijections Ψ and ψ of Theorem 3.1 and Proposition 3.1 provide a bijection

$$f : (\text{Ex}(\lambda, k) \cup B(\lambda, k)) \times \{k, \dots, n\} \rightarrow (\text{Ex}(\lambda', k) \cup B(\lambda', k)) \times \{k, \dots, n\}.$$

However, we have an additional bijection $g : B(\lambda, k) \rightarrow B(\lambda', k)$ given by $(H, T, i) \mapsto (H', T', i)$ at our disposal. Consider a sequence

$$(a, l) \xrightarrow{f} (b'_1, l'_1) \xrightarrow{g} (b_1, l_1) \xrightarrow{f} (b'_2, l'_2) \xrightarrow{g} (b_2, l_2) \xrightarrow{f} \dots$$

where $a \in \text{Ex}(\lambda, k)$, $b'_j \in B(\lambda', k)$ and $b_j \in B(\lambda, k)$. Since f and g are bijections, no pair (b'_j, l'_j) or (b_j, l_j) can recur. Since the considered sets are finite the sequence must end at some point with a pair (a', l') where $a' \in \text{Ex}(\lambda', k)$. Hence, by setting $\tilde{\psi}(a, l) := (a', l')$ we have found the desired bijection.

Acknowledgements

The author wants to thank Christian Krattenthaler and Cristoph Neumann for their helpful comments.

References

- [Ait43] Alexander Craig Aitken. The monomial expansion of determinantal symmetric functions. *Proc. Royal Soc. Edinburgh (A)* **61**, pages 300–310, 1943.
- [FRT54] J. S. Frame, G. de B. Robinson, and R. M. Thrall. The hook graphs of the symmetric group. *Canadian Journal of Mathematics* **6**, pages 316–325, 1954.
- [NPS97] Jean-Cristophe Novelli, Igor Pak, and Alexander V. Stoyanovskii. A direct bijective proof of the hook-length formula. *Discrete Mathematics and Theoretical Computer Science* **1**, pages 53–67, 1997.
- [NS13] Christoph Neumann and Robin Sulzgruber. A complexity theorem for the Novelli–Pak–Stoyanovskii-algorithm. Electronic preprint at arXiv:1306.5134v1, 2013.
- [Sag01] Bruce E. Sagan. *The Symmetric Group*. Springer, New York, 2nd edition, 2001.